

PRIMO ASSEGNAMENTO: BUBBLE

Informatica, Corso di Laurea in Fisica, Università di Pisa

AA 2019/20

1 Bubble Breaker

Il gioco Bubble Breaker si basa su una griglia, rappresentata da una matrice di caratteri di dimensione $L \times L$, dove ogni cella contiene un carattere (o colore), oppure è *vuota*.

Vi sono h colori in totale (è un input del gioco), rappresentati da i caratteri '0', '1' ... ' $h - 1$ ', che corrispondono alle caselle *piene*.

Ad esempio, se $h = 3$, i possibili colori corrispondono ai caratteri '0', '1', e '2'. Le caselle vuote contengono il carattere rappresentato dalla macro EMPTY in `bubble.h`.

Le regole del gioco sono le seguenti:

- La board del gioco è una matrice $L \times L$, dove la cella `board[0][0]` corrisponde alla cella in alto a sinistra.
- Una mossa corrisponde nello scegliere una cella `board[i][j]` *non vuota*, e far “esplodere” una porzione della matrice adiacente alla cella scelta, secondo le regole spiegate più avanti (funzioni `pop_lineare` e `pop_bubble`).
- Le celle della sezione esplosa diventano vuote (EMPTY); A questo punto si applica la gravità: gli elementi nelle celle che si trovano sopra caselle vuote cadono verso il basso fino ad appoggiarsi sulle caselle piene sottostanti, o sul fondo.
- Alla mossa vengono assegnati c^2 punti, dove c è il numero di caselle esplose tramite la mossa.
- E' anche possibile riempire nuovamente le celle vuote della board con valori casuali.

Vediamo un esempio: prendiamo una matrice 6×6 , con $h = 3$, e vediamo cosa succede scegliendo la casella 2,3 (evidenziata con parentesi nella prima immagine), e la direzione *ORIZ*.

0 0 0 0 0 0	0 0 0 0 0 0	0	0	0 2 0 1 2 0
0 0 0 2 0 0	0 0 0 2 0 0	0 0 0 0 0 0		0 0 0 0 0 0
0 1 1(1)1 0	0	0 0 0 2 0 0		0 0 0 2 0 0
2 2 0 2 2 0	2 2 0 2 2 0	2 2 0 2 2 0		2 2 0 2 2 0
1 0 0 0 2 0	1 0 0 0 2 0	1 0 0 0 2 0		1 0 0 0 2 0
1 0 0 0 0 0	1 0 0 0 0 0	1 0 0 0 0 0		1 0 0 0 0 0

Una board 6x6 (cella 2,3 tra parentesi)	Celle lasciate vuote da <code>pop_lineare(2,3,ORIZ)</code>	Effetto della gravità sulla board	possibile board dopo riempi(3)
--	---	--------------------------------------	-----------------------------------

La mossa esplode un segmento orizzontale corrispondente a 4 caselle di colore '1', e vale quindi 16 punti. Notiamo come le caselle sopra la sezione rimasta vuota cadono verso le righe inferiori per effetto delle gravità.

2 Regole di pop base

La regola di base per il pop è `pop_lineare`, da realizzare tramite la funzione `int pop_lineare (int i, int j, int d)`.

Data una casella ij non vuota, contenente il carattere $'c'$, e una direzione (ORIZ per orizzontale o VERT per verticale) `pop_lineare` esplode tutte le caselle nel segmento continuo di $'c'$ contenente la casella ij , dove il segmento è orizzontale o verticale a seconda della direzione specificata d , come in figura.

Dopo l'esplosione, la funzione deve anche applicare le regole di gravità. Alla fine, la funzione restituisce il punteggio assegnato alla mossa c^2 , dove c è il numero di caselle esplose dalla mossa; in caso di mossa non valida (colpo su una casella vuota, o coordinate non valide), la funzione restituisce invece il valore ERRORE (fornito in `bubble.h`)

Vediamo un esempio:

0 2 2 2 0 2	0 2 2 2 0 2	2 2 0 2	2 2 0 2	2 2 0 2
1 1 0 2 0 0	1 1 0 2 0 0	0 2 0 2 0 0	0 2 0 2 0 2	0 0 2 0 2
1 1 0 1 0 1	*(*)0 1 0 1	1 1 0 1 0 1	1 1 0 1 0 1	1 0 1 0 1
2 2 0 2 2 0	2 2 0 2 2 0	2 2 0 2 2 0	2 * 0 2 2 0	2 0 2 2 0
1 2 0 2 2 0	1 2 0 2 2 0	1 2 0 2 2 0	1(*)0 2 2 0	1 2 0 2 2 0
1 2 0 2 0 0	1 2 0 2 0 0	1 2 0 2 0 0	1 * 0 2 0 0	1 1 0 2 0 0
Una board 6x6	*: caselle esplose da <code>pop_lineare</code> (2,1,ORIZ)	La board dopo <code>pop_lineare</code> (2,1,ORIZ)	*: caselle esplose da <code>pop_lineare</code> (4,1,VERT)	La board dopo <code>pop_lineare</code> (4,1,VERT)

Le mosse fanno esplodere rispettivamente 2 e 3 caselle, e fanno quindi un punteggio di 4 e 9. Dopo aver eseguito queste due mosse, la mossa `pop_lineare(0,0,ORIZ)` non è valida in quanto la casella è vuota, quindi deve produrre come risultato ERRORE e non modificare la griglia.

3 Cosa deve essere realizzato

Lo studente deve realizzare un file `bubble.c` contenente le funzioni che servono ad eseguire alcuni componenti del gioco BUBBLE BREAKER.

I prototipi delle funzioni da realizzare in `bubble.c` si trovano nel file `bubble.h`.

- `void init_board(int h)`: inizializza ogni cella della board con un carattere nell'intervallo $'0', '1' \dots 'h - 1'$ scelto in modo casuale. **Importante:** il carattere che rappresenta un numero intero n può essere ottenuto tramite l'espressione $'0' + n$ (ad es, scrivendo `char x = '0' + 3` otteniamo che la variabile x ha come valore il carattere $'3'$)
- `void gravita()`: fa cadere gli elementi in caselle che si trovano sopra a caselle vuote verso il basso, fino ad appoggiarsi sulle caselle non vuote sottostanti, o sul fondo.
- `int pop_lineare (int i, int j, int d)`: esplode la cella `board[i][j]` e tutto il segmento lineare continuo (in verticale o orizzontale, a seconda del valore di d) con lo stesso carattere, come in figura sopra. Poi applica le regole di gravità. La funzione restituisce il punteggio della mossa (ovvero il quadrato del numero di celle esplose), o ERRORE in caso di mossa invalida (ad es, colpisco una casella vuota, o fuori dalla board).
- `void riempi (int h)`: Riempie tutte e sole le celle vuote della matrice con caratteri nell'intervallo $'0', '1' \dots 'h - 1'$.

I file `*test.c` contengono dei main che usano queste funzioni ed effettuano dei test sul loro funzionamento. Tali test possono essere attivati automaticamente utilizzando il `Makefile` come specificato nel file README. Solo il codice che supera con successo questi test può essere consegnato.

Tuttavia, è bene ricordare che il superamento dei test non garantisce la correttezza completa della soluzione, quindi invitiamo gli studenti ad analizzare attentamente i risultati ottenuti e le stampe effettuate prima della consegna.

4 Consegna

Lo studente deve consegnare un archivio contenente i file `bubble.c` e `gruppo.txt` (le istruzioni su come generare questo archivio sono nel file `README`).

Le funzioni realizzate devono essere adeguatamente commentate come discusso a lezione.

5 Regole di pop avanzate - FACOLTATIVO

Questa funzione non è obbligatoria per la consegna, ma può aggiungere fino a un altro punto di bonus se realizzata correttamente.

La regola avanzata per il pop è `pop_bubble`, da realizzare tramite la funzione `int pop_bubble (int i, int j)`.

La funzione esplode la cella `board[i][j]` e tutta la porzione raggiungibile dalla cella `ij` della matrice contenente lo stesso carattere. Poi applica le regole di gravità.

Ogni cella è considerata vicina alla cella alla sua sinistra, destra, sopra, e sotto, ma non diagonalmente. La figura sotto fa vedere un esempio.

La funzione restituisce il punteggio della mossa (ovvero il quadrato del numero di celle esplose), o ERRORE in caso di mossa invalida (ad es, colpisco una casella vuota, o fuori dalla board).

Esempio `pop_bubble`:

0 0 2 0 0 2	0 0 2 0 0 2	2 0 0 2	2 0 0 2	0 0 2
1 1 0 2 1 0	* * 0 2 1 0	0 2 1 0	(*)2 1 0	2 1 0
1 1 0 1 0 1	*(*)0 1 0 1	0 0 0 1 0 1	* * * 1 0 1	1 0 1
2 2 0 2 2 2	2 2 0 2 2 2	2 2 0 2 2 2	2 2 * 2 2 2	2 2 2 2
1 0 0 2 2 0	1 0 0 2 2 0	1 0 0 2 2 0	1 * * 2 2 0	1 2 2 2 0
1 2 0 2 0 0	1 2 0 2 0 0	1 2 0 2 0 0	1 2 * 2 0 0	1 2 2 2 0 0
Una board 6x6	*: caselle esplose da <code>pop_bubble</code> (2,1)	La board dopo <code>pop_bubble</code> (2,1)	*: caselle esplose da <code>pop_bubble</code> (1,2)	La board dopo <code>pop_bubble</code> (1,2)

I punteggi ottenuti in queste due chiamate sono rispettivamente 16 e 64.